[0268] reversing the graphical effect used to show the gloss text

[0269] change visibility of all lines of gloss text to invisible

[0270] change anchor underline style to solid

[0271] As those skilled in the art will understand, various software architectures are suitable for implementing spatial and temporal negotiations necessary for supporting the foregoing described examples of primary bodies of data presenting additional supporting annotations. For example, one contemplated negotiation architecture suitable for use in the negotiation module **36** of **FIG. 2** (and in the foregoing illustrated examples of this invention) fluidly takes into account the spatial extent and graphical presentation needs of both the primary and supporting bodies of data. (To more clearly describe them in terms of the architecture, primary and supporting data are sometimes referred to as "fluid objects, "primary or supporting objects" or "parent and child objects." in the following discussion). When a supporting object is called upon for examination, a structured negotiation session occurs between it and a primary object, resulting in a cooperating visual presentation of both. This process is a "negotiation" because neither participant in the exchange has complete control over the layout and presentation decision. The primary object cannot dictate how the supporting object is displayed, nor can the supporting object demand a particular region into which to expand itself. Instead, the participants make requests and proposals based on the information they have of their own layout and presentation needs. The sequence of requests and responses effectively exchange enough information so that a mutually satisfactory situation can be settled upon.

[0272] This "fluid" negotiation architecture enables a component model of fluid documents in which different fluid objects can co-exist without being designed specifically to interact with one another. A framework for a hierarchical graphical characteristics language allows fluid objects to gracefully handle requests from other fluid objects that present different or more complex graphical characteristics, determining which fluid objects will be affected, selecting and composing space-making strategies, ensuring salience of the expanded supporting object and protocols for multiple graphical domains, providing a negotiation sequence for resolving space allocation and presentation, and animating the expanding supporting object and the accommodating primary object.

[0273] When the user indicates interest in some supporting material (child object, a series of requests and responses are exchanged between the child object and the surrounding primary material (parent object. The sequence is as follows:

[0274] 1. Intent to expand. The child object contacts its parent, indicating that it would like to expand. It passes along information that describes graphical parameters that the child can control.

[0275] 2. Initial guidelines. The parent performs a preliminary determination of where the child object should expand and how it should present itself. It does this by considering the likely strategies it will employ to make space for the child. The parent then returns to the child guidelines for presentation.

[0276] 3. Request space. The child uses the guidelines to calculate how much space it will need when expanded. It is not required to follow the guidelines, but it will attempt to conform if possible. The space request, including the graphical characteristics that the child intends to conform to, is sent to the parent.

[0277] 4. Offer space. The parent once again considers its strategies for making space, starting with the strategy it most prefers. It tries each in turn, settling on a strategy (or composing multiple strategies) that best satisfies the child's request—the desired size if possible, or else somewhere within the range requested. The strategy may entail cascaded negotiations as the parent petitions its own parent for space.

[0278] 5. Consider offer. The child object considers the offered space and the graphical conditions attached to the offer. The child object selects a graphical presentation that conforms to the conditions and informs the parent how it chooses to accept the offer. The animation to the newly expanded state is then started.

[0279] The following sections examine each step in the negotiation by negotiation module **36** in detail:

[0280] Step 1: Intent to Expand

[0281] This step simply begins the negotiation. By passing along the graphical characteristics protocol that it adheres to (see next section), the child object informs the parent object about the type of graphical object it is, and thus what presentation capabilities it can accommodate.

[0282] Step 2: Initial Guidelines

[0283] In general, a child object does not know anything about its parent's presentation. Therefore, it does not initially know if its expanded form will present an effective contrast to the surrounding parent information. The parent object is in charge of communicating guidelines to the child object regarding desired graphical characteristics. For example, if the parent object is a page of text, using the font Times at 12 point, and colored black, it might suggest the following guidelines to an expanding textual annotation: sans-serif font between 10 and 14 point, highly saturated in color. This ensures that the expanded annotation is of comparable size, but is distinct with a different font and a bright color. Child objects try to conform to the guidelines if possible, but they are not constrained to following them. This is the first evidence of negotiation in the architecture—the child object asks its parent for guidelines in preparation for expansion; the parent makes recommendations; the child takes those recommendations under advisement as it prepares its first formal request for space.

[0284] The particular language for communicating a fluid object's graphical features, its graphical characteristics protocol, is specific to a class of fluid objects. For example, textual objects such as described in the previous paragraph use a simple protocol similar to the following:

[0285] font: {seriflsans-serif|Times|Helvetical . . . }

[0286] fontsize: {integer-range}

[0287] color: {dark|light|%gray|hue-range| . . . }